PHÂN TÍCH, THIẾT KẾ HƯỚNG ĐỐI TƯỢNG VÀ VAI TRÒ CỦA UML TRONG QUÁ TRÌNH PHÁT TRIỂN HỆ THỐNG PHẦN MỀM

Nguyễn Thị Phương Thủy
Khoa Công nghệ Thông tin, Trường Đại học Công nghệ Kỹ thuật Vinh
Email: phuongthuyskv@gmail.com

Tóm tắt: *Sự gia tăng về quy mô và mức độ phức tạp của các hệ thống phần mềm đặt ra yêu cầu về phương pháp phân tích, thiết kế phù hợp nhằm bảo đảm tính đúng đắn, khả năng mở rộng và bảo trì. Bài viết trình bày những vấn đề cơ bản của phân tích, thiết kế hệ thống, so sánh cách tiếp cận hướng chức năng với hướng đối tượng, đồng thời làm rõ vai trò của UML trong quá trình phát triển phần mềm. Phân tích, thiết kế hướng đối tượng tập trung vào các đối tượng, lớp và mối quan hệ trong miền nghiệp vụ, qua đó hỗ trợ đóng gói, che giấu thông tin, tái sử dụng và giảm sự phụ thuộc giữa các thành phần. UML cung cấp ngôn ngữ mô hình hóa trực quan giúp đặc tả yêu cầu, mô tả cấu trúc và hành vi của hệ thống, tạo cơ sở trao đổi giữa các thành viên trong quá trình phát triển. Việc kết hợp tư duy hướng đối tượng với UML góp phần nâng cao chất lượng phân tích, thiết kế và khả năng duy trì hệ thống phần mềm.*

Từ khóa: *Phân tích thiết kế hệ thống; hướng đối tượng; UML; hệ thống thông tin; phát triển phần mềm.*

OBJECT-ORIENTED ANALYSIS AND DESIGN AND THE ROLE OF UML IN SOFTWARE SYSTEM DEVELOPMENT

Abstract: *The increasing scale and complexity of software systems have created a growing need for appropriate analysis and design approaches to ensure correctness, scalability, and maintainability. This paper presents fundamental aspects of system analysis and design, compares functional and object-oriented approaches, and examines the role of the Unified Modeling Language (UML) in software development. Object-oriented analysis and design focuses on objects, classes, and relationships within the application domain, thereby supporting encapsulation, information hiding, reuse, and reduced dependencies among components. UML provides a visual modeling language for specifying requirements and representing the structure and behavior of software systems, thereby facilitating communication and collaboration among development team members. The integration of object-oriented thinking and UML contributes to improving the quality of system analysis and design as well as the maintainability of software systems.*

Keywords: *System analysis and design; object-oriented approach; UML; information systems; software development.*

Nhận bài: 20/09/2026 Phản biện: 22/10/2026 Duyệt đăng: 26/10/2026

I. ĐẶT VẤN ĐỀ

Phần mềm ngày càng được sử dụng rộng rãi trong quản lý, giáo dục, thương mại, sản xuất và nhiều lĩnh vực khác. Khi quy mô hệ thống tăng, yêu cầu về tính đúng đắn, khả năng mở rộng, bảo trì và thích ứng với thay đổi trở nên quan trọng. Vì vậy, phát triển phần mềm không chỉ là hoạt động lập trình mà cần một quá trình phân tích và thiết kế có hệ thống trước khi xây dựng sản phẩm.

Phân tích hệ thống nhằm xác định vấn đề, yêu cầu, các đối tượng tham gia và mối quan hệ trong hệ thống; thiết kế chuyển kết quả phân tích thành cấu trúc và cách thức vận hành có thể triển khai. Một mô hình phù hợp giúp hạn chế sự không rõ ràng của yêu cầu, phát hiện sớm những vấn đề trong thiết kế và tạo cơ sở cho các giai đoạn lập trình, kiểm thử, bảo trì (Bộ Giáo dục và Đào tạo, 2020; Nguyễn Văn Ba, 2012; Nguyễn Kim Anh, 2018).

Trong quá trình phát triển phần mềm, cách tiếp cận hướng chức năng từng được sử dụng phổ biến với việc phân rã hệ thống thành các chức năng và thủ tục xử lý. Tuy nhiên, đối với những hệ thống có nhiều thực thể nghiệp vụ, dữ liệu liên kết và yêu cầu thường xuyên thay đổi, hướng đối tượng cung cấp cách tiếp cận phù hợp hơn khi mô hình hóa hệ thống theo các đối tượng và mối quan hệ của chúng (Trần Đức Minh & Lê Văn Ngọc, 2015). Cùng với đó, UML trở thành công cụ mô hình hóa trực quan hỗ trợ biểu diễn yêu cầu, cấu trúc và hành vi của hệ thống (Đặng Văn Đức, 2016).

II. CƠ SỞ VỀ HỆ THỐNG THÔNG TIN

Hệ thống có thể được hiểu là tập hợp các thành phần được tổ chức và liên kết với nhau nhằm đạt một mục tiêu chung. Khi nghiên cứu một hệ thống cần xác định các thành phần, thuộc tính, mối quan hệ, phạm vi và môi trường tác động. Việc xác định ranh giới giúp làm rõ phạm vi của hệ thống và những thành phần thuộc hoặc không thuộc đối tượng phân tích (Nguyễn Văn Ba, 2012; Nguyễn Kim Anh, 2018).

Hệ thống thông tin quản lý là sự kết hợp giữa con người, phần cứng, phần mềm, dữ liệu và các quy trình nhằm thu thập, xử lý, lưu trữ và cung cấp thông tin phục vụ hoạt động quản lý. Dữ liệu đầu vào có thể phát sinh từ bên trong hoặc bên

ngoài tổ chức; quá trình xử lý bao gồm phân tích, tính toán, cập nhật và lưu trữ; kết quả được phân phối đến các đối tượng sử dụng theo những nhu cầu khác nhau. Chất lượng thông tin được thể hiện qua tính kịp thời, chính xác, đầy đủ và phù hợp với mục đích sử dụng (Bộ Giáo dục và Đào tạo, 2020; Nguyễn Văn Ba, 2012; Nguyễn Kim Anh, 2018).

Do đó, phân tích và thiết kế hệ thống thông tin phải đồng thời xem xét dữ liệu, hoạt động nghiệp vụ, người sử dụng, các tác nhân bên ngoài và mối quan hệ giữa chúng. Đây là cơ sở để lựa chọn thành phần, chức năng và mô hình phù hợp trước khi triển khai hệ thống.

III. PHÂN TÍCH, THIẾT KẾ HƯỚNG CHỨC NĂNG VÀ HƯỚNG ĐỐI TƯỢNG

Phân tích, thiết kế hướng chức năng tập trung vào các chức năng mà hệ thống phải thực hiện. Hệ thống thường được phân rã từ tổng thể thành các chức năng và thủ tục nhỏ hơn theo hướng từ trên xuống. Cách tiếp cận này phù hợp với những bài toán có quy trình xử lý tương đối rõ ràng và ít thay đổi. Tuy nhiên, trong hệ thống lớn, nhiều chức năng có thể cùng sử dụng dữ liệu nên sự phụ thuộc giữa các thành phần tăng. Khi cấu trúc dữ liệu hoặc yêu cầu thay đổi, nhiều chức năng liên quan có thể phải điều chỉnh, gây khó khăn cho bảo trì và mở rộng (Bộ Giáo dục và Đào tạo, 2020; Trần Đức Minh & Lê Văn Ngọc, 2015).

Phân tích, thiết kế hướng đối tượng tập trung vào các đối tượng tồn tại trong miền nghiệp vụ, lớp, thuộc tính, phương thức và mối quan hệ giữa chúng. Chẳng hạn, trong hệ thống thư viện có thể xác định các đối tượng như Sách, Tạp chí và Độc giả; mỗi đối tượng có đặc điểm và hành vi riêng, đồng thời tương tác với các đối tượng khác. Cách tiếp cận này đưa cấu trúc phần mềm gần hơn với các thực thể của bài toán thực tế (Trần Đức Minh & Lê Văn Ngọc, 2015).

Hướng đối tượng khai thác các đặc trưng như đóng gói, che giấu thông tin, kế thừa và tái sử dụng. Việc tổ chức trách nhiệm cho các đối tượng phù hợp giúp giảm sự phụ thuộc giữa các thành phần, thuận lợi cho mở rộng và bảo trì (Trần Đức Minh & Lê Văn Ngọc, 2015).

Hướng chức năng và hướng đối tượng đều nhằm giải quyết bài toán phân tích và thiết kế nhưng khác nhau về trọng tâm. Hướng chức năng quan tâm chủ yếu đến hệ thống thực hiện những công việc gì và các chức năng được phân rã như thế nào; hướng đối tượng quan tâm đến hệ thống gồm những đối tượng nào, mỗi đối tượng có trách nhiệm gì và chúng tương tác ra sao.

Việc lựa chọn phụ thuộc vào đặc điểm bài toán, quy mô, mức độ thay đổi và khả năng mở rộng trong tương lai. Đối với các hệ thống có nhiều thực thể nghiệp vụ, quan hệ phức tạp và yêu cầu phát triển lâu dài, hướng đối tượng có ưu thế trong việc tổ chức và tái sử dụng các thành phần phần mềm (Bộ Giáo dục và Đào tạo, 2020; Trần Đức Minh & Lê Văn Ngọc, 2015).

IV. QUY TRÌNH PHÁT TRIỂN PHẦN MỀM VÀ VAI TRÒ CỦA UML

Phát triển phần mềm thường trải qua các hoạt động chính gồm xác định yêu cầu, phân tích, thiết kế, lập trình, kiểm thử và vận hành, bảo trì. Các hoạt động này có mối liên hệ chặt chẽ và có thể được thực hiện lặp lại tùy theo mô hình phát triển. Yêu cầu là cơ sở xác định hệ thống cần cung cấp chức năng gì; phân tích làm rõ bài toán và các thành phần của miền nghiệp vụ; thiết kế xác định cách tổ chức các thành phần; lập trình hiện thực thiết kế; kiểm thử đánh giá sản phẩm; bảo trì thực hiện sửa lỗi, cải tiến và thích ứng với yêu cầu mới (Bộ Giáo dục và Đào tạo, 2020; Nguyễn Văn Ba, 2012).

Trong quá trình này, mô hình hóa giúp chuyển những yêu cầu và ý tưởng trừu tượng thành các biểu diễn có cấu trúc. UML (Unified Modeling Language) là ngôn ngữ mô hình hóa được sử dụng để đặc tả, trực quan hóa và tài liệu hóa hệ thống. UML hỗ trợ mô tả cả cấu trúc tĩnh và hành vi động của hệ thống, qua đó tạo sự thống nhất trong trao đổi giữa các bên tham gia phát triển (Đặng Văn Đức, 2016).

UML không thay thế quá trình phân tích và thiết kế mà đóng vai trò là công cụ mô hình hóa. Hiệu quả của UML phụ thuộc vào việc xác định đúng yêu cầu, lựa chọn mô hình phù hợp và sử dụng mô hình đúng mục đích (Đặng Văn Đức, 2016).

Trong phân tích và thiết kế hướng đối tượng, một số biểu đồ UML thường được sử dụng gồm: Biểu đồ ca sử dụng (Use Case Diagram): mô tả các chức năng hệ thống cung cấp và mối quan hệ giữa tác nhân với hệ thống, hỗ trợ xác định yêu cầu; Biểu đồ lớp (Class Diagram): mô tả cấu trúc tĩnh của hệ thống thông qua lớp, thuộc tính, phương thức và các mối quan hệ; Biểu đồ tuần tự (Sequence Diagram): mô tả trình tự trao đổi thông điệp giữa các đối tượng theo thời gian; Biểu đồ trạng thái (State Diagram): mô tả các trạng thái và sự chuyển đổi trạng thái của đối tượng; Biểu đồ thành phần (Component Diagram): mô tả các thành phần phần mềm và quan hệ phụ thuộc giữa chúng.

Các biểu đồ có thể được sử dụng kết hợp thay vì tách rời. Chẳng hạn, ca sử dụng giúp xác định yêu cầu; từ yêu cầu có thể xác định các đối tượng và lớp; biểu đồ lớp mô tả cấu trúc; biểu đồ tuần tự làm rõ cách các đối tượng tương tác để thực hiện chức năng. Cách tiếp cận này tạo sự liên kết từ yêu cầu đến thiết kế và hiện thực hệ thống (Đặng Văn Đức, 2016).

Nhờ biểu diễn trực quan, UML giúp giảm sự mơ hồ trong trao đổi yêu cầu, hỗ trợ làm việc nhóm, phát hiện sớm những điểm chưa hợp lý trong thiết kế và tạo tài liệu phục vụ bảo trì. Tuy nhiên, việc xây dựng quá nhiều mô hình không cần thiết có thể làm tăng chi phí và gây khó khăn cho quá trình phát triển. Vì vậy, cần lựa chọn những mô hình thực sự phục vụ mục tiêu phân tích, thiết kế và giao tiếp trong dự án (Đặng Văn Đức, 2016).

V. BÀN LUẬN VÀ Ý NGHĨA TRONG PHÁT TRIỂN PHẦN MỀM

Chất lượng của hệ thống phần mềm chịu ảnh hưởng đáng kể từ chất lượng của quá trình phân tích và thiết kế. Nếu yêu cầu không được xác định đầy đủ hoặc các thành phần của hệ thống được tổ chức không hợp lý, những vấn đề phát sinh có thể kéo dài sang lập trình, kiểm thử và bảo trì. Ngược lại, mô hình phù hợp giúp nhóm phát triển có cách nhìn thống nhất về hệ thống trước khi triển khai.

Hướng đối tượng tạo điều kiện tổ chức phần mềm theo các thực thể và trách nhiệm của chúng, qua đó hỗ trợ đóng gói, tái sử dụng và mở rộng. UML bổ sung khả năng mô hình hóa trực quan, giúp chuyển các khái niệm nghiệp vụ thành những biểu diễn có thể trao đổi và kiểm tra. Sự kết hợp giữa tư duy hướng đối tượng và UML vì vậy có ý nghĩa trong các hệ thống như quản lý sinh viên, thư viện, bán hàng, quản lý đào tạo và nhiều hệ thống thông tin nghiệp vụ khác (Đặng Văn Đức, 2016; Trần Đức Minh & Lê Văn Ngọc, 2015).

Trong đào tạo công nghệ thông tin, nội dung phân tích, thiết kế hướng đối tượng và UML không chỉ cung cấp kiến thức về mô hình hóa mà còn góp phần hình thành năng lực phân tích bài toán, xác định yêu cầu, thiết kế cấu trúc, giải quyết vấn đề và giao tiếp trong phát triển phần mềm. Việc gắn các khái niệm với những hệ thống thực tế giúp người học chuyển từ việc hiểu ký hiệu UML sang vận dụng mô hình để giải quyết bài toán cụ thể.

VI. KẾT LUẬN

Phân tích và thiết kế là những hoạt động quan trọng trong quá trình phát triển hệ thống phần mềm. So với cách tiếp cận tập trung vào chức năng, hướng đối tượng tổ chức hệ thống dựa trên các đối tượng, lớp và mối quan hệ, đồng thời khai thác đóng gói, che giấu thông tin, kế thừa và tái sử dụng. Cách tiếp cận này phù hợp với những hệ thống có cấu trúc phức tạp và yêu cầu thường xuyên thay đổi (Trần Đức Minh & Lê Văn Ngọc, 2015).

UML cung cấp ngôn ngữ mô hình hóa thống nhất để mô tả yêu cầu, cấu trúc và hành vi của hệ thống thông qua nhiều loại biểu đồ. Việc sử dụng UML một cách phù hợp giúp tăng khả năng giao tiếp, kiểm tra thiết kế và duy trì tài liệu trong quá trình phát triển (Đặng Văn Đức, 2016). Do đó, kết hợp phân tích, thiết kế hướng đối tượng với UML không chỉ có giá trị trong thực tiễn phát triển phần mềm mà còn có ý nghĩa trong đào tạo nguồn nhân lực công nghệ thông tin.

TÀI LIỆU THAM KHẢO

Bộ Giáo dục và Đào tạo. (2020). Giáo trình phân tích và thiết kế hệ thống thông tin [Giáo trình dùng cho các cơ sở đào tạo đại học ngành Công nghệ thông tin].

Đặng Văn Đức. (2016). UML và mô hình hóa hệ thống thông tin. Nhà xuất bản Khoa học và Kỹ thuật.

Nguyễn Kim Anh. (2018). Phân tích và thiết kế hệ thống thông tin quản lý. Nhà xuất bản Giáo dục Việt Nam.

Nguyễn Văn Ba. (2012). Phân tích và thiết kế hệ thống thông tin. Nhà xuất bản Đại học Quốc gia Hà Nội.

Trần Đức Minh & Lê Văn Ngọc. (2015). Phân tích thiết kế hệ thống thông tin – Hướng cấu trúc và hướng đối tượng. Nhà xuất bản Thống kê.